IN THE CLAIMS

Please amend claims 1, 7, 15, 21, and 23 and cancel claims 2, 3, 8, 9, 16, 17, 24 & 25, as follows:

CLAIMS

1. (Currently Amended) A method of lexicographically sorting data, comprising:

receiving a set of N cyclic shifts of N characters identifiable by an array of indexes {0, 1, 2, ..., N-1};

sorting the set of cyclic shifts based on a comparison of a first character of each cyclic shift;

for an nth sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, ..., up to $2^n >$ N:

(A) identifying a subset of the N cyclic shifts with equivalent $2^{(n-1)}$ characters;

(B) sorting ~~at least a~~ the subset of the N cyclic shifts ~~which are identifiable by a subset array of indexes in the array in accordance with a previous sort of cyclic shifts associated with the subset array of indexes plus $2^{(n-1)}$*modulo(N)~~ with equivalent $2^{(n-1)}$ characters by: (i) identifying an index value of the first character in each of the subset of the N cyclic shifts; (ii) adding $2^{(n-1)}$*modulo(N) to the identified index value; (iii) sorting the next $2^{(n-1)}$ index values in each of the subset of the N cyclic shifts; and (iv) reordering the subset of N cyclic shifts based upon the sorting step (iii) so as to lexicographically sort the first $2^{(n)}$ characters in the subset of N cyclic shifts; and

(C) repeating the sorting for a next nth sorting iteration as necessary until all of the characters in the set of N cyclic shifts are lexicographically sorted.

2. (Canceled)

3. (Canceled)

4. (Original) The method of claim 1, wherein the act of sorting based on the comparison of the first character comprises a radix sort.

5. (Original) The method of claim 1, wherein the method is included in a data compression algorithm.

6. (Original) The method of claim 1, wherein the method is included in a Burrows-Wheeler Transform (BWT) clustering procedure.

7. (Currently Amended) A computer program product, comprising:

memory;

computer instructions stored in the memory; and

the computer instructions being executable by a processor for lexicographically sorting cyclic data by:

receiving a set of N cyclic shifts of N characters identifiable by an array of indexes {0, 1, 2, ..., N-1};

sorting the set of cyclic shifts based on a comparison of a first character of each cyclic shift;

for an nth sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, ..., up to $2^n > N$:

(A) identifying a subset of the N cyclic shifts with equivalent $2^{(n-1)}$ characters;

(B) sorting ~~at least a~~ the subset of the N cyclic shifts ~~which are identifiable by a subset array of indexes in the array in accordance with a previous sort of cyclic shifts associated with the subset array of indexes plus $2^{(n-1)}$*modulo(N)~~ with equivalent $2^{(n-1)}$ characters by: (i) identifying an index value of the first character in each of the subset of the N cyclic shifts; (ii) adding $2^{(n-1)}$*modulo(N) to the identified index value; (iii) sorting the next $2^{(n-1)}$ index values in each of the subset of the N cyclic shifts; and (iv) reordering the subset of N cyclic shifts based upon the sorting step (iii) so as to lexicographically sort the first $2^{(n)}$ characters in the subset of N cyclic shifts; and

(C) repeating the sorting for a next nth sorting iteration as necessary until all of the characters in the set of N cyclic shifts are lexicographically sorted.

8. (Canceled)

9. (Canceled)

10. (Original) The computer program product of claim 7, wherein the act of sorting based on the comparison of the first character comprises a radix sort.

11. (Original) The computer program product of claim 7, wherein the computer instructions are included in a data compression algorithm.

12. (Original) The computer program product of claim 7, wherein the computer instructions are included in a Burrows-Wheeler Transform (BWT) clustering procedure.

13. (Original) The computer program product of claim 7, wherein a computer embodies the computer program product.

14. (Original) The computer program product of claim 7, wherein a mobile communication device embodies the computer program product.

15. (Currently Amended) A system for communicating data, comprising:

a wireless packet data network;

a mobile communication device which operates in the wireless packet data network;

a computer coupled to the wireless packet data network;

the computer including a computer readable memory;

computer instructions stored in the computer readable memory;

the computer instructions being executable by a processor for performing a data compression algorithm to generate compressed data which is communicated to the mobile communication device, the data compression algorithm including a sorting algorithm for lexicographically sorting cyclic data by:

receiving a set of N cyclic shifts of N characters identifiable by an array of indexes {0,1, 2, …, N-1};

sorting the set of cyclic shifts based on a comparison of a first character of each cyclic shift;

for an n^{th} sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, …, up to $2^n > N$:

(A) identifying a subset of the N cyclic shifts with equivalent $2^{(n-1)}$ characters;

(B) sorting ~~at least a~~ the subset of the N cyclic shifts ~~which are identifiable by a subset array of indexes in the array in accordance with a previous sort of cyclic shifts associated with the subset array of indexes plus $2^{(n-1)}$*modulo(N)~~ with equivalent $2^{(n-1)}$ characters by: (i) identifying an index value of the first character in each of the subset of the N cyclic shifts; (ii) adding $2^{(n-1)}$*modulo(N) to the identified index value; (iii) sorting the next $2^{(n-1)}$ index values in each of the subset of the N cyclic shifts; and (iv) reordering the subset of N cyclic shifts based upon the sorting step (iii) so as to lexicographically sort the first $2^{(n)}$ characters in the subset of N cyclic shifts; and

(C) repeating the sorting for a next n^{th} sorting iteration as necessary until all of the characters in the set of N cyclic shifts are lexicographically sorted.

16. (Canceled)

17. (Canceled)

18. (Original) The system of claim 15, wherein the act of sorting based on the comparison of the first character comprises a radix sort.

19. (Original) The system of claim 15, wherein the computer instructions are included in a data compression algorithm.

20. (Original) The system of claim 15, wherein the computer instructions are included in a Burrows-Wheeler Transform (BWT) clustering procedure.

21. (Currently Amended) A method of lexicographically sorting data, comprising:

receiving a plurality of N cyclic shifts of N characters to be sorted;

providing a sorting array of N elements {0, 1, 2, ..., N-1}, each element being an index which uniquely corresponds to one cyclic shift of the plurality of N cyclic shifts;

initially sorting the plurality of N cyclic shifts of N characters based on a first character of each cyclic shift of N characters, and ordering the indexes in the sorting array in accordance with such sorting;

iteratively sorting the initially sorted plurality of N cyclic shifts by:

for an nth sorting iteration of the set of cyclic shifts, where n = 1, 2, 3, ..., up to $2^n > N$:

(A) identifying a subset of the N cyclic shifts with equivalent $2^{(n-1)}$ characters;

(B) sorting ~~at least a~~ the subset of the N cyclic shifts ~~which are identifiable by a subset array of indexes in the array in accordance with a previous sort of cyclic shifts associated with the subset array of indexes plus $2^{(n-1)}$*modulo(N)~~ with equivalent $2^{(n-1)}$ characters by: (i) identifying an index value of the first character in each of the subset of the N cyclic shifts; (ii) adding $2^{(n-1)}$*modulo(N) to the identified index value; (iii) sorting the next $2^{(n-1)}$ index values in each of the subset of the N cyclic shifts; and (iv) reordering the subset of N cyclic shifts based upon the sorting step (iii) so as to lexicographically sort the first $2^{(n)}$ characters in the subset of N cyclic shifts; and

(C) repeating the sorting for a next nth sorting iteration as necessary until all of the characters in the set of N cyclic shifts are lexicographically sorted.

22. (Original) The method of claim 21, further comprising:

providing an inverse sorting array of N elements, each element identifying a sorted position for the cyclic shift corresponding to a position of the element in the inverse sorting array; and

utilizing the inverse sorting array for sorting the subset array of indexes.

23. (Currently Amended) The method of claim 21, further comprising:

providing an equivalents array of N elements for identifying the subset ~~array~~ of N cyclic shifts ~~indexes~~ corresponding to cyclic shifts having equivalent characters.

24. (Canceled)

25. (Canceled)